PATENT ABSTRACTS OF JAPAN

(11)Publication number : **10-133927**

(43)Date of publication of application : **22.05.1998**

(51)Int.Cl. G06F 12/00

(21)Application number : **09-232930**

(22)Date of filing : **28.08.1997**

(71)Applicant : **TOSHIBA CORP**

(72)Inventor : **HIRAYAMA HIDEAKI**
SHIROKIBARA TOSHIO

(30)Priority

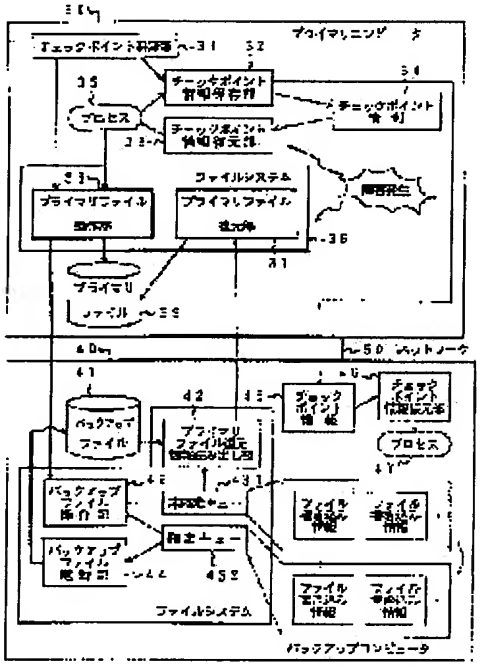
Priority number : **08233021** Priority date : **03.09.1996** Priority country : **JP**

(54) COMPUTER SYSTEM AND FILE MANAGING METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a computer system realizing rolling back at the time of generating a fault without waiting the saving of data before update at the time of updating a file.

SOLUTION: When writing, etc., is requested to the file, 'file writing information' is preserved in an unidentified queue 431 to instantly update only a primary file 39. Then after a check point is picked up, 'file writing information' preserved in the queue 431 is moved to an identified queue 432 to reflect to a backup file 41. On the other hand, at the time of recovery, all the pieces of data before update corresponding to data updated after a finally picked check point are read from the file 41 based on 'file writing information' preserved in the queue 431 to recover the file 39 to the point of a check point time by using this read data before update.



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平10－133927

(43)公開日　平成10年(1998)５月22日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　12/00	５３１	Ｃ０６Ｆ　12/00	５３１Ｒ
			５３１Ｍ

審査請求　未請求　請求項の数16　ＯＬ　（全 15 頁）

(21)出願番号　特願平9－232930

(22)出願日　平成９年(1997)８月28日

(31)優先権主張番号　特願平8－233021

(32)優先日　平８(1996)９月３日

(33)優先権主張国　日本（ＪＰ）

(71)出願人　000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地

(72)発明者　平山　秀昭
東京都青梅市末広町２丁目９番地　株式会社東芝青梅工場内

(72)発明者　白木原　敏雄
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内

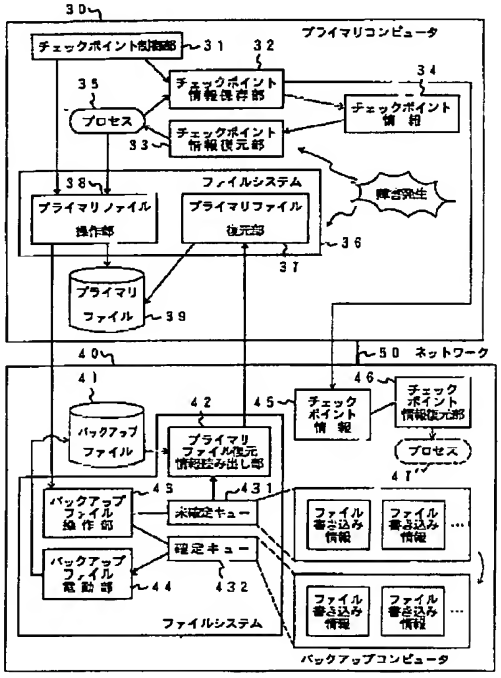
(74)代理人　弁理士　鈴江　武彦　（外６名）

(54)【発明の名称】　コンピュータシステムおよびファイル管理方法

(57)【要約】

【課題】ファイルを更新する際に更新前データの退避を待機することなく障害発生時のロールバックを実現するコンピュータシステムを提供する。

【解決手段】ファイルに対して書き込みなどが要求された際、その「ファイル書き込み情報」を未確定キュー４３１に保存してプライマリファイル３９のみを即座に更新する。そして、チェックポイントが採取された後に、未確定キュー４３１に保存された「ファイル書き込み情報」を確定キュー４３２に移動させてバックアップファイル４１への反映を行なう。一方、リカバリを行なうときには、未確定キュー４３１に保存された「ファイル書き込み情報」に基づき、最後に採取したチェックポイント以降に更新されたデータに対応する更新前のデータをバックアップファイル４１からすべて読み出し、この読み出した更新前のデータを用いてプライマリファイル３９をチェックポイント時点に復元する。



【特許請求の範囲】

【請求項１】 運用系および待機系の２つのコンピュータで２重化されたコンピュータシステムであって、中断された処理を再開始するためのチェックポイントを定期的に採取し、前記運用系および待機系双方のコンピュータ上に保存するコンピュータシステムにおいて、

前記運用系のコンピュータ上で実行されるプロセスによって更新されるファイルを前記運用系および待機系双方のコンピュータで２重化して設けておき、

前記プロセスからファイルの更新が指示されたときに、その更新情報を前記待機系のコンピュータ上に保存して運用系のファイルのみを更新し、その更新が完了した時点でその更新の要求元に対し更新完了を通知する手段と、

前記チェックポイントが採取された後に、前記更新情報に示される更新内容を前記待機系のファイルに反映させる手段とを具備してなることを特徴とするコンピュータシステム。

【請求項２】 前記更新情報を前記運用系のコンピュータ上にバッファリングしておき、前記チェックポイントの採取時点までに前記待機系のコンピュータに一括転送する手段をさらに具備してなることを特徴とする請求項１記載のコンピュータシステム。

【請求項３】 前記プロセスがアボートしたときに、最後のチェックポイント以降に実行されたファイルの更新に対する更新前のデータを前記更新情報により前記待機系のファイルから読み出し、前記運用系のファイルを前記チェックポイント時点の状態に復元した後、前記プロセスを前記チェックポイントから再実行する手段をさらに具備してなることを特徴とする請求項１または２記載のコンピュータシステム。

【請求項４】 前記プロセスがアボートしたときに、最後のチェックポイント以降に保存された更新情報を削除し、前記チェックポイント以前の更新情報により示される更新を前記待機系のファイルに反映させた後、前記プロセスを前記待機系のコンピュータ上で前記チェックポイントから再実行する手段をさらに具備してなることを特徴とする請求項１または２記載のコンピュータシステム。

【請求項５】 前記運用系のコンピュータまたはこの運用系のコンピュータを制御するオペレーティングシステムに障害が発生したときに、最後のチェックポイント以降に保存された更新情報を削除し、前記チェックポイント以前の更新情報により示される更新を前記待機系のファイルに反映させた後、前記プロセスを前記待機系のコンピュータ上で前記チェックポイントから再実行する手段をさらに具備してなることを特徴とする請求項１または２記載のコンピュータシステム。

【請求項６】 前記待機系のコンピュータまたはこの待機系のコンピュータを制御するオペレーティングシステムに障害が発生したときに、前記チェックポイントおよび更新情報の待機系のコンピュータへの転送を停止する手段をさらに具備してなることを特徴とする請求項１または２記載のコンピュータシステム。

【請求項７】 前記運用系のファイルに障害が発生したときに、最後のチェックポイント以降に保存された更新情報を削除し、前記チェックポイント以前の更新情報により示される更新を前記待機系のファイルに反映させた後、前記プロセスを前記待機系のコンピュータ上で前記チェックポイントから再実行する手段をさらに具備してなることを特徴とする請求項１または２記載のコンピュータシステム。

【請求項８】 前記待機系のファイルに障害が発生したときに、前記チェックポイントおよび更新情報の待機系のコンピュータへの転送を停止する手段をさらに具備してなることを特徴とする請求項１または２記載のコンピュータシステム。

【請求項９】 待機系のファイルの切り離しが行なわれたときに、第３のコンピュータ上に新たに待機系のファイルを確保する手段をさらに具備してなることを特徴とする請求項１または２記載のコンピュータシステム。

【請求項１０】 待機系のファイルを用いて前記プロセスの前記チェックポイントからの再実行が行なわれたときに、前記待機系のファイルを運用系に切り替えて、前記運用系のコンピュータ上に新たに待機系のファイルを確保する手段をさらに具備してなることを特徴とする請求項１または２記載のコンピュータシステム。

【請求項１１】 運用系および待機系の２つのコンピュータで２重化され、中断された処理を再開始するためのチェックポイントを定期的に採取して前記運用系および待機系双方のコンピュータ上に保存し、前記運用系のコンピュータ上で実行されるプロセスによって更新されるファイルを前記運用系および待機系双方のコンピュータ上で２重化して設けたコンピュータシステムのファイル管理方法において、

前記プロセスからファイルの更新が指示されたときに、その更新情報を前記待機系のコンピュータ上に保存して運用系のファイルのみを更新し、その更新が完了した時点でその更新の要求元に対し更新完了を通知するステップと、

前記チェックポイントが採取された後に、前記更新情報に示される更新内容を前記待機系のファイルに反映させるステップとを具備してなることを特徴とするファイル管理方法。

【請求項１２】 最後のチェックポイント以降に実行されたファイルの更新に対する更新前のデータを前記更新情報により前記待機系のファイルから読み出し、前記運用系のファイルを前記チェックポイント時点の状態に復元した後、前記プロセスを前記チェックポイントから再実行するステップをさらに備えたことを特徴とする請求

項１１記載のファイル管理方法。
【請求項１３】　最後のチェックポイント以降に保存された更新情報を削除し、前記チェックポイント以前の更新情報により示される更新を前記待機系のファイルに反映させた後、前記プロセスを前記待機系のコンピュータ上で前記チェックポイントから再実行するステップをさらに備えたことを特徴とする請求項１１記載のファイル管理方法。
【請求項１４】　運用系および待機系の２つのコンピュータで２重化され、中断された処理を再開始するためのチェックポイントを定期的に採取して前記運用系および待機系双方のコンピュータ上に保存し、前記運用系のコンピュータ上で実行されるプロセスによって更新されるファイルを前記運用系および待機系双方のコンピュータ上で多重化して設けたコンピュータシステムのファイルを管理するためのプログラムであって、
前記プロセスからファイルの更新が指示されたときに、その更新情報を前記待機系のコンピュータ上に保存して運用系のファイルのみを更新し、その更新が完了した時点でその更新の要求元に対して更新完了を通知し、
前記チェックポイントが採取された後に、前記更新情報に示される更新内容を前記待機系のファイルに反映させるように前記コンピュータを動作させるためのプログラムを格納したコンピュータ読取可能な記憶媒体。
【請求項１５】　前記プログラムは、最後のチェックポイント以降に実行されたファイルの更新に対する更新前のデータを前記更新情報により前記待機系のファイルから読み出し、前記運用系のファイルを前記チェックポイント時点の状態に復元した後、前記プロセスを前記チェックポイントから再実行するように前記コンピュータをさらに動作させる請求項１４記載のコンピュータ読取可能な記憶媒体。
【請求項１６】　前記プログラムは、最後のチェックポイント以降に保存された更新情報を削除し、前記チェックポイント以前の更新情報により示される更新を前記待機系のファイルに反映させた後、前記プロセスを前記待機系のコンピュータ上で前記チェックポイントから再実行するように前記コンピュータをさらに動作させる請求項１４記載のコンピュータ読取可能な記憶媒体。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】この発明は、たとえばネットワーク接続された複数のコンピュータにより構成されるネットワークコンピューティング環境などにおいて、高い信頼性を必要とするグループコンピューティング処理、データベース処理、およびトランザクション処理などに適用して好適なコンピュータシステムおよびファイル管理方法に関する。
【０００２】
【従来の技術】ＣＰＵによって実行されるプロセスのアドレス空間やコンテクスト、およびファイルなどの状態を定期的に採取して（これをチェックポイントと称す）、障害が発生したときに、最後に採取したチェックポイントの状態を復元し、その時点からプロセスの実行を再開始するといった障害からの回復機能を有したシステムにおいては、従来より外部入出力処理に関して課題があった。すなわち、障害が発生したときに、最後に採取したチェックポイントからプロセスを再実行させる際、プロセスのアドレス空間やプロセッサのコンテクストなどの状態は復元できるが、外部入力装置の状態の復元は容易ではなかった。
【０００３】たとえば、ファイルに対する書き込みをキャンセルすることは困難であるために、ファイルに対して書き込みを行なうときには、データをファイルに書き込む前に書き込み以前のデータを事前に読み込んで保存を行ない、その後にファイルへのデータ書き込みを行なっていた。
【０００４】図１５は、ファイルに対する書き込みをキャンセルすることが困難なため、ファイルに対して書き込みを行なうときに、データをファイルに書き込む前に書き込み以前のデータを事前に読み込んで保存を行ない、その後にファイルへのデータ書き込みを行なう従来のシステムの仕組みを説明する図である。
【０００５】この例では、時刻ｔ１のチェックポイントを採取した時点において“ＡＢＣＤ”の４バイトのデータからなるファイルに、時刻ｔ２において１バイト目に“Ｘ”をｗｒｉｔｅしている（１）。この場合、従来では、ファイルの１バイト目に“Ｘ”をｗｒｉｔｅする前に、ファイルの１バイト目のデータ“Ｂ”をｒｅａｄしておき（これをｕｎｄｏログとも言う）（２）、その後でファイルの１バイト目に“Ｘ”をｗｒｉｔｅしている（３）。
【０００６】その後、時刻ｔ３において障害が発生したために、プロセスを最後に採取されたチェックポイントの状態（ｔ１）にロールバックする。ファイルは、チェックポイントｔ１以降に１バイト目が“Ｘ”に更新されているが、更新時に採取されたｕｎｄｏログを用いることにより、チェックポイントｔ１の状態を復元している。なお、このｕｎｄｏログは、次のチェックポイント時に不要となり廃棄される。
【０００７】また、たとえば２つのコンピュータにより構築され、その一方（プライマリコンピュータ）を運用系、他方（バックアップコンピュータ）を待機系として振り分けて２重化し、プライマリコンピュータに障害が発生したときに、バックアップコンピュータが処理を引き継くことによってシステムの可用性を高めるといったシステムも存在する。そして、このようなシステムで、前述したようにチェックポイントを定期的に採取していけば、信頼性をさらに向上させることが可能となる。
【０００８】

【発明が解決しようとする課題】この様に、プロセスのアドレス空間やコンテクスト、およびファイルなどの状態、すなわち、チェックポイントを定期的に採取していき、障害が発生したときに、最後に採取したチェックポイントの状態を復元し、その時点からプロセスの実行を再開始するといった障害からの回復機能を有したシステム（２重化されているかどうかを問わない）においては、その信頼性は向上されるが、一方で、ファイルの更新（たとえば書き込み）を行なうときに、一旦更新前のデータをファイルから読み込んで、それからファイルへの更新を行なわなければならなかったために、ファイルの更新性能を低下させるという課題があった。

【０００９】この発明は、このような実情に鑑みてなされたものであり、チェックポイントを定期的に採取して、障害が発生したときには最後に採取したチェックポイントの状況を復元し、その時点からプロセスの実行を再開始するといった障害からの回復機能を有したシステムにおいて、ファイルの更新を行なうときに、更新前のデータをファイルから読み込むなどといったことを不要とし、ファイルの更新性能を大幅に改善することを可能とするコンピュータシステムおよびファイル管理方法を提供することを目的とする。

【００１０】

【課題を解決するための手段】この発明のコンピュータシステムは、運用系および待機系の２つのコンピュータで２重化されたコンピュータシステムであって、中断された処理を再開始するためのチェックポイントを定期的に採取し、前記運用系および待機系双方のコンピュータ上に保存するコンピュータシステムにおいて、前記運用系のコンピュータ上で実行されるプロセスによって更新されるファイルを前記運用系および待機系双方のコンピュータで２重化して設けておき、前記プロセスからファイルの更新が指示されたときに、その更新情報を前記待機系のコンピュータ上に保存して運用系のファイルのみを更新し、その更新が完了した時点でその更新の要求元に対し更新完了を通知する手段と、前記チェックポイントが採取された後に、前記更新情報に示される更新内容を前記待機系のファイルに反映させる手段とを具備してなることを特徴とする。

【００１１】この発明のコンピュータシステムにおいては、プロセスがファイルの更新を要求したときに、その更新内容を示す更新情報を取得して保存するとともに、運用系のコンピュータに配置されたファイル（運用系ファイル）のみを即座に更新して、その結果を要求元であるプロセスに返答する。そして、チェックポイントが採取された後に、その保存しておいた更新情報で示される更新内容を、待機系のコンピュータに配置されたファイル（待機系ファイル）に反映させる。

【００１２】一方、たとえばプロセスがアボートしたときなどには、保存しておいた更新情報に基づいて、最後に採取したチェックポイント以降に更新されたデータに対応する更新前のデータを待機系ファイルからすべて読み出し、この読み出した更新前のデータを用いて運用系ファイルをチェックポイント時点に復元する。

【００１３】すなわち、このコンピュータシステムにおいては、従来のようにファイルを更新するときに、更新前のデータを読み出して退避させておくといった処理の完了を通常処理に待機させることなく、障害時のファイルのリカバリが実現されることになり、信頼性を損なうことなくファイルの更新性能を飛躍的に向上させることが可能となる。

【００１４】また、運用系ファイルの復元に代えて、最終のチェックポイント以前に保存された更新情報で示される更新内容すべてが反映された待機系ファイルを用いたチェックポイントからのプロセスの再実行も有効である。すなわち、運用系のコンピュータの障害などにより、運用系ファイルを用いての再開始が不可能な場合などにおける処理の継続も確保されることになり、システムの可用性を向上させることになる。また、この場合には、第３のコンピュータに新たに待機系ファイルを確保すれば、システムの可用性をさらに向上させることが可能となる。

【００１５】

【発明の実施の形態】まず、図１を参照してこの発明の基本原理を説明する。図１に示すように、この発明のコンピュータシステムは、運用系システム１０と待機系システム２０とで多重化されたシステムを前提とする。以下にそれぞれの動作を説明する。

【００１６】（通常処理）

（１）運用系システム１０でアプリケーションプログラム１１がＷｒｉｔｅシステムコールを発行する。

【００１７】（２）ジャケットルーチン１２がＷｒｉｔｅシステムコールをフックし、運用系のオペレーティングシステムにＷｒｉｔｅシステムコールを発行するとともに、そのＷｒｉｔｅ要求を待機系システム２０に送信する。ただし、待機系システム２０に即座にＷｒｉｔｅ要求を送信する必要はなく、次のチェックポイントまでに送信すればよい。また、待機系システム２０では、受信したＷｒｉｔｅ要求を即座に実行するのではなく、一旦未確定キュー２１１に格納する。

【００１８】（３）チェックポイント処理が指示されると、運用系システム１０は、溜っているＷｒｉｔｅ要求をすべて待機系システム２０に送信し終えなければならない。

【００１９】（４）一方、待機系システム２０では、未確定キュー２１１に格納されたＷｒｉｔｅ要求を確定キュー２１２に移動する。

【００２０】（５）この確定キュー２１２に移されたＷｒｉｔｅ要求は、待機系システム２０のオペレーティングシステムによって順次処理されていく。

【００２１】すなわち、通常処理において発生するファイル更新においては、更新前のデータを読み出して退避させておくといった処理の完了を待機することがない。

【００２２】（ロールバック処理）

（３）˝障害が発生したようなときに、運用系システム１０および待機系システム２０の双方にロールバック処理が指示される。

【００２３】（４）˝このとき、運用系システム１０に残存するＷｒｉｔｅ要求を、すべて待機系システム２０に送信する。また、待機系の未確定キュー２１１に格納されたＷｒｉｔｅ要求は、最後のチェックポイント以降に発行されたものであるので、逆にこれを参照して待機系ファイル２３から更新前のデータを読み出し、この読み出した更新前のデータを用いて運用系ファイル１４をロールバックする。これにより、運用系ファイル１４および待機系ファイル２３の双方のファイルが最後のチェックポイント時点の状態になる。

【００２４】（５）˝そして、待機系システム２０は、未確定キュー２１１に残存するＷｒｉｔｅ要求をすべてキャンセルする。

【００２５】これにより、チェックポイント時点からの再開始が可能となる。

【００２６】次に、この発明の実施の形態を説明する。

【００２７】（第１の実施形態）まず、この発明の第１の実施形態を説明する。図２にはこの発明の第１の実施形態に係るコンピュータシステムのシステム構成が示されている。図２に示したように、本実施形態のコンピュータシステムは、コンピュータがプライマリコンピュータ３０と、バックアップコンピュータ４０とで２重化されており、これらはネットワーク５０で接続されている。このプライマリコンピュータ３０とバックアップコンピュータ４０とは、前述した運用系システム１０および待機系システム２０双方をそれぞれに備えており、いずれかで運用系システム１０が動作するときに、他方では待機系システム２０が動作する。ここでは、プライマリコンピュータ３０側で運用系システム１０、バックアップコンピュータ４０側で待機系システム２０をそれぞれ説明する。

【００２８】プロセス３５は、プライマリコンピュータ３０上で実行され、プライマリファイル３９とバックアップファイル４１とで２重化されたファイルを更新する。ここで、プライマリファイル３９はプライマリコンピュータ３０上に、バックアップファイル４１はバックアップコンピュータ４０上に配置され、プライマリコンピュータ３０上のファイルシステム３６およびバックアップコンピュータ４０上のファイルシステム４８を介して更新される。

【００２９】プライマリコンピュータ３０上のファイルシステム３６は、プライマリファイル操作部３８とプライマリファイル復元部３７とを含んでいる。一方、バックアップコンピュータ４０上のファイルシステム４８は、バックアップファイル操作部４３、未確定キュー４３１、確定キュー４３２、バックアップファイル更新部４４およびプライマリファイル復元情報読み出し部４２を含んでいる。

【００３０】プロセス３５がこの２重化されたファイルを更新する場合、プライマリファイル操作部３８およびバックアップファイル操作部４３を経由して行なう。プロセス３５がこの２重化されたファイルに対応するｗｒｉｔｅを行なうと、プライマリファイル３９は、そのまま即座に更新されるが、バックアップファイル４１はその時点では更新されずに、「ファイル書き込み情報」がバックアップファイル操作部４３を経由して、バックアップコンピュータ４０上の未確定キュー４３１に保存される。

【００３１】また、プロセス３５がチェックポイントを採取する場合には、チェックポイント制御部３１が、チェックポイント情報保存部３２とプライマリファイル操作部３８とにその指示を出す。チェックポイント情報保存部３２は、チェックポイント採取の指示を受け取ると、チェックポイント情報（アドレス空間とプロセッサコンテクスト）をプライマリコンピュータ３０上およびバックアップコンピュータ４０上に保存する（プライマリコンピュータ３０上のチェックポイント情報３４およびバックアップコンピュータ４０上のチェックポイント情報４５）。

【００３２】一方、プライマリファイル操作部３８は、チェックポイント採取の指示を受け取ると、バックアップファイル操作部４３を経由して、未確定キュー４３１に保存されていた「ファイル書き込み情報」を確定キュー４３２に移動させる。この確定キュー４３２に移動された「ファイル書き込み情報」は、チェックポイント採取後に、バックアップファイル更新部４４によってバックアップファイル４１の更新のために使用され、バックアップファイル４１の更新後に廃棄される。これにより、チェックポイント以降にプライマリファイル３９に対して行なわれたものと同じｗｒｉｔｅ操作が、バックアップファイル４１に対しても行なわれることになる。

【００３３】プロセス３５がアボートなどの障害を発生させ、プロセス３５をプライマリコンピュータ３０上で最後に採取したチェックポイントから再実行する場合、アドレス空間とプロセッサコンテクストとは、プライマリコンピュータ３０上のチェックポイント情報復元部３７によって復元される。

【００３４】ファイルに関しては、バックアップファイル４１は、チェックポイント以降の更新は未だ未確定キュー４３１に「ファイル書き込み情報」が保存されているだけであり、実際には更新されていないので復元は不要である。しかしながら、プライマリファイル３９は、チェックポイント以降にすでに更新が行なわれているの

で復元が必要である。したがって、未確定キュー４３１に保存された「ファイル書き込み情報」に基づき、プライマリファイル３９の更新前データをバックアップファイル４１からｒｅａｄし、このｒｅａｄした更新前データをプライマリファイル３９にｗｒｉｔｅすることによって復元する。そして、この後、未確定キュー４３１に保存された「ファイル書き込み情報」を廃棄する。なお、確定キュー４３２に「ファイル書き込み情報」が保存されている場合には、その「ファイル書き込み情報」のバックアップファイル４１への反映が完了した後に、前述した復元処理を開始する。

【００３５】一方、プライマリコンピュータ３０またはプライマリコンピュータ３０を制御するオペレーティングシステムがシステムダウンなどの障害を発生させ、プロセス３５をバックアップコンピュータ４０上で最後に採取したチェックポイントから再実行する場合には、アドレス空間とプロセッサコンテクストとは、チェックポイント情報復元部４６によってプロセス４７に復元される。

【００３６】ファイルに関しては、バックアップファイル４１は、チェックポイント以降の更新は未だ未確定キュー４３１に「ファイル書き込み情報」が保存されているだけであり、実際には更新されていないので復元は不要である。

【００３７】なお、この「ファイル書き込み情報」のプライマリコンピュータ３０からバックアップコンピュータ４０への転送については最適化が可能である。障害が発生したときに、プライマリコンピュータ３０がダウンしなかった場合は、プライマリファイル３９を復元し、プライマリファイル３９を用いてチェックポイントからの処理を再開する。一方、障害が発生したときに、プライマリコンピュータ３０がダウンした場合には、バックアップファイル４１を用いてチェックポイントから処理を再開する。

【００３８】それゆえに、「ファイル書き込み情報」は、プライマリファイル操作部３８からバックアップファイル操作部４３に即時に送る必要はない。すなわち、これらの「ファイル書き込み情報」は、次のチェックポイントまでに送ればよいので、転送効率を考慮すると、一旦プライマリファイル操作部３８において蓄積しておき、「一定容量蓄積された」、「一定時間経過した」および「チェックポイント採取が要求された」といった事象の発生をトリガとして、バックアップファイル操作部４３にまとめて送るということが可能である。

【００３９】図３には、本実施形態を適用するコンピュータシステムの概略構成が示されている。コンピュータはプライマリコンピュータ３０とバックアップコンピュータ４０とで２重化されており、プライマリコンピュータ３０にはディスク装置６０ａが、バックアップコンピュータ４０にはディスク装置６０ｂがそれぞれ接続されている。プロセス３５はプライマリコンピュータ上で実行され、また、このプロセス３５がアクセスするファイルは、プライマリファイル３９とバックアップファイル４１とで２重化されており、各々ディスク装置６０ａとディスク装置６０ｂとに配置されている。

【００４０】そして、チェックポイントは、チェックポイント情報をプライマリコンピュータ３０側（プライマリチェックポイント情報３４）と、バックアップコンピュータ４０側（バックアップチェックポイント情報４５）の両方に保持する。なお、この図では、チェックポイントをディスク装置上に保持しているが、メモリ上に保持しても構わない。

【００４１】もし、プライマリコンピュータ３０またはプライマリコンピュータ３０を制御するオペレーティングシステムにシステムダウンなどの障害が発生した場合には、バックアップコンピュータ４０側でチェックポイント情報４５を用いてプロセス４７を再実行する。この場合プロセス４７は、バックアップファイル４１を使用することになる。

【００４２】また、プライマリファイル３９またはバックアップファイル４１を複数個持ち、３重化以上のファイルシステムを作ることも可能である。この場合、たとえば３重化ファイルシステムならば、

（１）２個のプライマリファイルと１個のバックアップファイル

（２）１個のプライマリファイルと２個のバックアップファイル

といった組み合わせが考えられる。

【００４３】図４は、本実施形態においてファイルを更新する様子を示す図である。この例では、プライマリコンピュータ３０上で動くプロセス３５が、４バイトのデータ“ＡＢＣＤ”を持つ２重化されたファイル（プライマリコンピュータ３０上のプライマリファイル３９と、バックアップコンピュータ４０上のバックアップファイル４１）に対し、時刻ｔ１において１バイト目に“Ｘ”をｗｒｉｔｅしている（１）。これによってプライマリファイル３９は即時に更新されるが、バックアップファイル４１は即時には更新されずに、「ファイル書き込み情報」のみを保存している。

【００４４】この後、時刻ｔ２においてチェックポイントが採取されることによって、先程の「ファイル書き込み情報」の実行が確定する（２）。そして時刻ｔ２以降で、確定された「ファイル書き込み情報」に基づいて、バックアップファイル４１の更新を実行している。

【００４５】図５は、本実施形態において障害発生時にプライマリファイルを復元する様子を示す図である。この例では、プライマリコンピュータ３０上で動くプロセス３５が、４バイトのデータ“ＡＢＣＤ”を持つ２重化されたファイル（プライマリコンピュータ３０上のプライマリファイル３９と、バックアップコンピュータ４０

上のバックアップファイル４１）に対し、時刻ｔ１において１バイト目に“X”をｗｒｉｔｅしている（１）。これによってプライマリファイル３９は即時に更新されるが、バックアップファイル４１は即時には更新されずに、「ファイル書き込み情報」のみを保存している。

【００４６】この後、時刻ｔ２において障害が発生している（２）。すなわち、時刻ｔ１おける「ファイル書き込み情報」でプライマリファイル３９は更新されているため復元の必要があるが、バックアップファイル４１は未だ更新されていないため復元の必要がない。ここで時刻ｔ１において保存された「ファイル書き込み情報」によって、プライマリファイル３９の更新部分がかわる。そこで、プライマリファイル３９の復元においては、未確定の「ファイル書き込み情報」に示された位置のデータをバックアップファイル４１からｒｅａｄし、そのｒｅａｄしたデータをプライマリファイル３９にｗｒｉｔｅすることによって、プライマリファイル３９を復元する。

【００４７】そして、プライマリコンピュータ３０上で取られているチェックポイントを用いて、プライマリコンピュータ３０上でプロセス３５を再実行している。この再実行されたプロセス３５は、復元されたプライマリファイル３９を使用する。

【００４８】図6は、ファイル操作部が「ファイル書き込み」を指示されたときの処理の流れを示すフローチャートである。この場合、まず、「ファイル書き込み情報」を保存し、未確定キュー４３１にリンクする（ステップＡ１）。次に、「ファイル書き込み情報」にしたがって、プライマリファイル３９の更新を行なう（ステップＡ２）。この時点で、「ファイル書き込み」操作は完了したとして、要求側に完了通知を行なう（ステップＡ３）。

【００４９】図7は、ファイル操作部が「チェックポイント採取」を指示されたときの処理の流れを示すフローチャートである。この場合、保存されている「ファイル書き込み情報」を未確定キュー４３１から確定キュー４３２に移動する（ステップＢ１）。

【００５０】図8は、バックアップファイル更新部の処理の流れを示すフローチャートである。この場合、まず、確定キュー４３２に「ファイル書き込み情報」がリンクされているかどうかを検査する（ステップＣ１）。もし、リンクされていない場合（ステップＣ１のＮ）、バックアップファイル更新部４４は、この検査を続行する。一方、リンクされている場合には（ステップＣ１のＹ）、確定キュー４３２にリンクされている「ファイル書き込み情報」に基いて、バックアップファイル４１を更新する（ステップＣ２）。そして、実行した「ファイル書き込み情報」を確定キュー４３２からはずす（ステップＣ３）。

【００５１】図9は、プロセス３５にアボートなどの障害が発生し、プロセス３５をプライマリコンピュータ３０上で最後に採取したチェックポイントから再実行する場合の処理の流れを示すフローチャートである。

【００５２】プロセス３５に障害が発生すると、まず、プライマリコンピュータ３０上のチェックポイント情報復元部３３に、「アドレス空間とプロセッサコンテクストとの復元を指示する（ステップＤ１）。次に、プライマリファイル復元部３３に、「プライマリファイルの復元」を指示する（ステップＤ２）。

【００５３】図１０は、プライマリコンピュータ３０上のチェックポイント情報復元部が「アドレス空間とプロセッサコンテクストの復元」を指示された場合の処理の流れを示すフローチャートである。この場合、まず、プロセス３５のアドレス空間を復元する（ステップＥ１）。次に、プロセス３５のチェックポイント採取時のプロセッサコンテクストの状態を復元する（ステップＥ２）。

【００５４】図１１は、プライマリファイル復元部３７が、「プライマリファイルの復元」を指示された場合の処理の流れを示すフローチャートである。この場合、まず、未確定キュー４３１に、「ファイル書き込み情報」がリンクされているかどうかを検査する（ステップＦ１）。「ファイル書き込み情報」がリンクされている場合には（ステップＦ１のＹ）未確定キュー４３１にリンクされている「ファイル書き込み情報」にしたがって、プライマリファイル３９の中の更新されている部分のデータをバックアップファイル４１からｒｅａｄし、そのＲｅａｄしたデータをプライマリファイル３９にｗｒｉｔｅすることにより、プライマリファイル３９のその更新されている部分のデータを復元する（ステップＦ２）。そして、復元に利用した「ファイル書き込み情報」を、未確定キュー４３１からはずす（廃棄する）（ステップＦ３）。この処理は、未確定キュー４３１にリンクた「ファイル書き込み情報」が無くなるまで繰り返される。

【００５５】プライマリコンピュータ３０またはプライマリコンピュータ３０を制御するオペレーティングシステムにシステムダウンなどの障害が発生した場合には、プロセス３５をバックアップコンピュータ４０上で最後に採取したチェックポイントから再実行する。この場合は、バックアップファイル４１で処理を引き継ぐ。図１２は、障害が発生したときに、バックアップファイル４１で処理を引き継ぐ様子を示す図である。

【００５６】この例では、プライマリコンピュータ３０上で動作するプロセス３５が、４バイトのデータ“ＡＢＣＤ”を持つ２重化されたファイル（プライマリコンピュータ３０上のプライマリファイル３９と、バックアップコンピュータ４０上のバックアップファイル４１）に対し、時刻ｔ１において１バイト目に“X”をｗｒｉｔｅしている（１）。これによってプライマリファイル３

９は即時に更新されるが、バックアップファイル４１は即時には更新されずに、「ファイル書き込み情報」のみを保存している。

【００５７】この後、時刻ｔ２においてプライマリコンピュータ３０に障害が発生している（２）。この場合、バックアップコンピュータ４０上に取られたチェックポイントを用いて、バックアップコンピュータ４０上でプロセス４７を再実行している。このとき、プロセス４７は、バックアップファイル４１を用いて処理を継続するわけだが、時刻ｔ１においてプライマリファイル３９は更新されているが、バックアップファイル４１は未だ更新されていないので、バックアップコンピュータ４０上でのプロセス４７の再実行においては、バックアップファイル４２がそのまま使用できる。

【００５８】なお、障害発生によりバックアップファイルを切り離した場合には、その後に新たなバックアップファイルを作成することによって、再び図１の様な初期状態を再現することができ、再度の障害発生に対しても回復処理が可能となる。

【００５９】また、障害発生によってバックアップファイルで処理を引き継ぎ、チェックポイントから処理を再実行した場合には、その後、バックアップファイルをプライマリファイルとして新たなバックアップファイルを作成することにより、再び図１の様な初期状態を再現することができ、再度の障害発生に対しても回復処理が可能となる。この再度バックアップファイルを作成する場合には、以下の様な２つの方法がある。

【００６０】（１）バックアップファイル切り離し後のプライマリファイルの更新情報とデータとを保存しておき、バックアップファイルを再接続する場合には、バックアップファイルに前記切り離し後のプライマリファイルの更新情報とデータとを反映させる。

【００６１】（２）プライマリファイルをバックアップファイルにコピーする。ただし、コピー中にもプライマリファイルが更新され続けている場合には、コピーを始めると同時にファイルの更新情報とデータとをバックアップファイルにも反映させる。

【００６２】さらに、この２つの方法を組み合わせた以下の様な方法も有効である。

【００６３】（３）切り離されたバックアップファイル（あるいは障害発生前のプライマリファイル）を再接続することを前提に、一定時間が経過するまでは（１）の方法が取れる様に、バックアップファイル切り離し後のプライマリファイルの更新情報とデータとを保存しておく。一定時間を経過したら、（１）の方法は締め、バックアップファイル切り離し後のプライマリファイルの更新情報とデータとの保存は止めて、（２）の方法を取るようにする。また、切り離されたバックアップファイル以外のファイルで再接続する場合にも、バックアップファイル切り離し後のプライマリファイルの更新情報とデータとの保存は止めて、（２）の方法を取る。

【００６４】（第２の実施形態）次に、この発明の第２の実施形態を説明する。第１の実施形態では、２重化されたコンピュータシステムを説明したが、この発明は、２重化されていないコンピュータ上のファイルシステムに適用しても効果がある。そこで、本実施形態では、２重化されていないコンピュータ上のファイルシステムに適用した場合を例に説明する。図１３は、この発明を２重化されていないコンピュータ上のファイルシステムに適用した場合の構成図である。このシステムでは、コンピュータは２重化されておらず、コンピュータ３０のみが存在する。プロセス３５は、このコンピュータ３０上で実行され、プライマリファイル３９とバックアップファイル４１とで２重化されたファイルを更新する。すなわち、これらプライマリファイル３９およびバックアップファイル４１は、共にコンピュータ３０上に配置され、ファイルシステム３６を介して更新される。

【００６５】コンピュータ３０上のファイルシステム３６は、プライマリファイル操作部３８、プライマリファイル復元部３７、バックアップファイル操作部４３、未確定キュー４３１、確定キュー４３２、バックアップファイル更新部４４およびプライマリファイル復元情報読み出し部４２を含んでいる。

【００６６】プロセス３５がこの２重化されたファイルを更新するときは、プライマリファイル操作部３８およびバックアップファイル操作部４３を経由して行なう。プロセス３５がこの２重化されたファイルに対するｗｒｉｔｅを行なうと、プライマリファイル３９はそのまま更新されるが、バックアップファイル４１は更新されずに、「ファイル書き込み情報」がバックアップファイル操作部４３を経由して未確定キュー４３１に保存される。

【００６７】また、プロセス３５がチェックポイントを採取するときには、チェックポイント制御部３１が、チェックポイント情報保存部３２とプライマリファイル操作部４３に指示を出す。チェックポイント情報保存部３２はチェックポイント採取の指示を受けると、アドレス空間とプロセッサコンテクストとをコンピュータ３０上に行なう（チェックポイント情報３４）。

【００６８】一方、プライマリファイル操作部３８は、チェックポイント採取の指示を受けると、バックアップファイル操作部４３を経由して、未確定キュー４３１に保存されていた「ファイル書き込み情報」を確定キュー４３２に移動させる。確定キュー４３２に移動された「ファイル書き込み情報」は、チェックポイント採取後に、バックアップファイル更新部４４によってバックアップファイル４１の更新のために使用され、バックアップファイル４１の更新後に廃棄される。これにより、チェックポイント以降にプライマリファイル３９に対して行なわれたのと同じように、ｗｒｉｔｅ操作がバックア

ップファイル４１に対して行なわれる。

【００６９】プロセス３５にアボートなどの障害が発生し、プロセス３５をコンピュータ３０上で最後に採取したチェックポイントから再実行する場合、アドレス空間とプロセッサコンテクストは、コンピュータ３０上のチェックポイント情報復元部３３によって復元される。

【００７０】ファイルに関しては、バックアップファイル４１は、チェックポイント以降の更新が未だ未確定キュー４３１に「ファイル書き込み情報」が保存されているだけであり、実際には更新されていないので復元は不要である。しかしながら、プライマリファイル３９は、チェックポイント以降にすでに更新が行なわれているので復元が必要である。したがって、未確定キュー４３１に保存された「ファイル書き込み情報」に基づき、プライマリファイル３９の更新前データをバックアップファイル４１からｒｅａｄし、このＲｅａｄした更新前データをプライマリファイル３９にｗｒｉｔｅすることによって復元する。そして、この後、未確定キュー４３１に保存された「ファイル書き込み情報」を廃棄する。なお、確定キュー４３２に「ファイル書き込み情報」が保存されている場合には、その「ファイル書き込み情報」のバックアップファイル４１への反映が完了した後に、前述した復元処理を開始する。

【００７１】図１４には、本実施形態を適用するコンピュータシステムの概略構成が示されている。本実施形態のシステムはコンピュータ３０のみで稼働し２重化されていない。コンピュータ３０にはディスク装置６０ａとディスク装置６０ｂとが接続されている。プロセス３５はコンピュータ３０上で実行され、また、このプロセス３５がアクセスするファイルは、プライマリファイル３９とバックアップファイル４１とで２重化されており、各々ディスク装置６０ａとディスク装置６０ｂとに配置されている。

【００７２】このように、この発明を適用することにより、プロセスのアドレス空間やプロセッサのコンテクストなどの状態（チェックポイント情報）を定期的に保存しながら実行を続け、障害が発生したときには最後に保存したチェックポイントからプロセスを再実行させることによる障害時対策を施したシステムにおいて、ファイルの更新を行なう際に、一旦更新前データをファイルから読み込む必要がなくなるため、ファイルの更新性能が大幅に改善される。

【００７３】なお、前述の実施形態に記載したファイルの管理方法は、コンピュータに実行させることのできるプログラムとしてフロッピィディスク、光ディスクおよび半導体メモリなどの記録媒体に格納して頒布することが可能である。

【００７４】

【発明の効果】以上詳述したように、この発明によれば、プロセスがファイルの更新を要求したときに、その更新内容を示す更新情報を取得して保存するとともにプライマリファイルのみを即座に更新し、チェックポイントが採取された後に、その保存しておいた更新情報で示される更新内容をバックアップファイルに反映させる。そして、たとえばプロセスがアボートしたときなどには、保存しておいた更新情報に基づいて、最後に採取したチェックポイント以降に更新されたデータに対応する更新前のデータをバックアップファイルからすべて読み出し、この読み出した更新前のデータを用いてプライマリファイルをチェックポイント時点に復元し、プロセスの再実行を開始する（バックアップファイルを用いたプロセスの再実行の開始も可能）。

【００７５】すなわち、このコンピュータシステムにおいては、従来のようにファイルを更新するときに、更新前のデータを読み出して退避させておくといった処理の完了を通常処理に待機させることなく、障害時のファイルのリカバリが実現されることになり、信頼性を損なうことなくファイルの更新性能を飛躍的に向上させることが可能となる。

【図面の簡単な説明】

【図１】この発明の基本原理を説明するための概念図。

【図２】この発明の第１の実施形態に係るコンピュータシステムのシステム構成を示す図。

【図３】同実施形態を適用するコンピュータシステムの概略構成を示す図。

【図４】同実施形態においてファイルを更新する様子を示す図。

【図５】同実施形態において障害発生時にプライマリファイルを復元する様子を示す図。

【図６】同実施形態のファイル操作部が「ファイル書き込み」を指示されたときの処理の流れを示すフローチャート。

【図７】同実施形態のファイル操作部が「チェックポイント採取」を指示されたときの処理の流れを示すフローチャート。

【図８】同実施形態のバックアップファイル更新部の処理の流れを示すフローチャート。

【図９】同実施形態のプロセスにアボートなどの障害が発生し、プロセスをプライマリコンピュータ３０上で最後に採取したチェックポイントから再実行する場合の処理の流れを示すフローチャート。

【図１０】同実施形態のプライマリコンピュータ上のチェックポイント情報復元部が「アドレス空間とプロセッサコンテクストとの復元」を指示された場合の処理の流れを示すフローチャート。

【図１１】同実施形態のプライマリファイル復元部が「プライマリファイルの復元」を指示された場合の処理の流れを示すフローチャート。

【図１２】同実施形態の障害が発生したときにバックアップファイルで処理を引き継ぐ様子を示す図。

【図１３】この発明の第２の実施形態に係るコンピュータシステムのシステム構成を示す図。

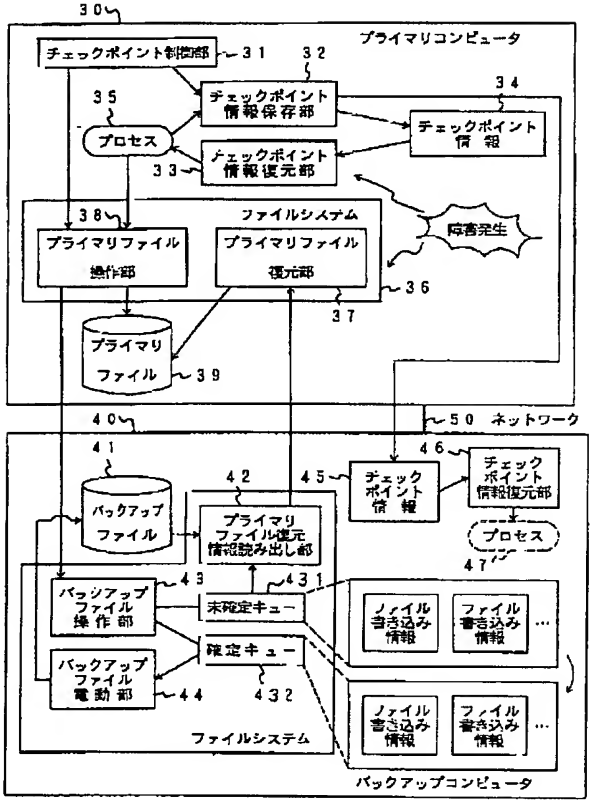
【図１４】同実施形態を適用するコンピュータシステムの概略構成を示す図。

【図１５】従前のファイルに対する書き込みをキャンセルすることが困難なため、ファイルに対して書き込みを行なうときに、データをファイルに書き込む前に書き込み以前のデータを事前に読み込んで保存を行ない、その後にファイルへのデータ書き込みを行なう従来のシステムの仕組みを説明する図。

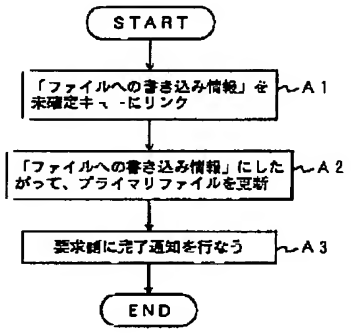
【符号の説明】

１０…運用系システム、１１…アプリケーションプログラム、１２…ジャケットルーチン、１３…ＯＳバッファキャッシュ、１４…ディスク装置、２０…待機系システム、２１デーモン、２１１…未確定キュー、２１２…確定キュー、２２…ＯＳバッファキャッシュ、２３…ディスク装置、３０…プライマリコンピュータ、３１…チェックポイント制御部、３２…チェックポイント情報保存部、３３…チェックポイント情報復元部、３４…チェックポイント情報、３５…プロセス、３６…ファイルシステム、３７…プライマリファイル復元部、３８…プライマリファイル操作部、３９…プライマリファイル、４０…バックアップコンピュータ、４１…バックアップファイル、４２…プライマリファイル復元情報読み出し部、４３…バックアップファイル操作部、４３１…未確定キュー、４３２…確定キュー、４４…バックアップファイル更新部、４５…チェックポイント情報、４６…チェックポイント情報復元部、４７…プロセス、５０…ネットワーク、６０ａ，６０ｂ…ディスク装置。

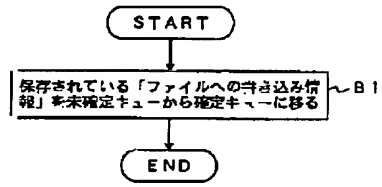
【図２】



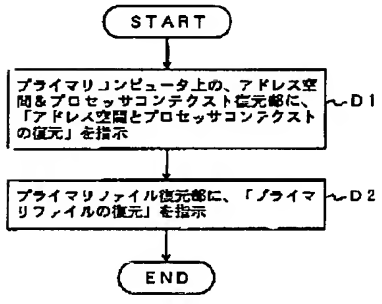
【図６】



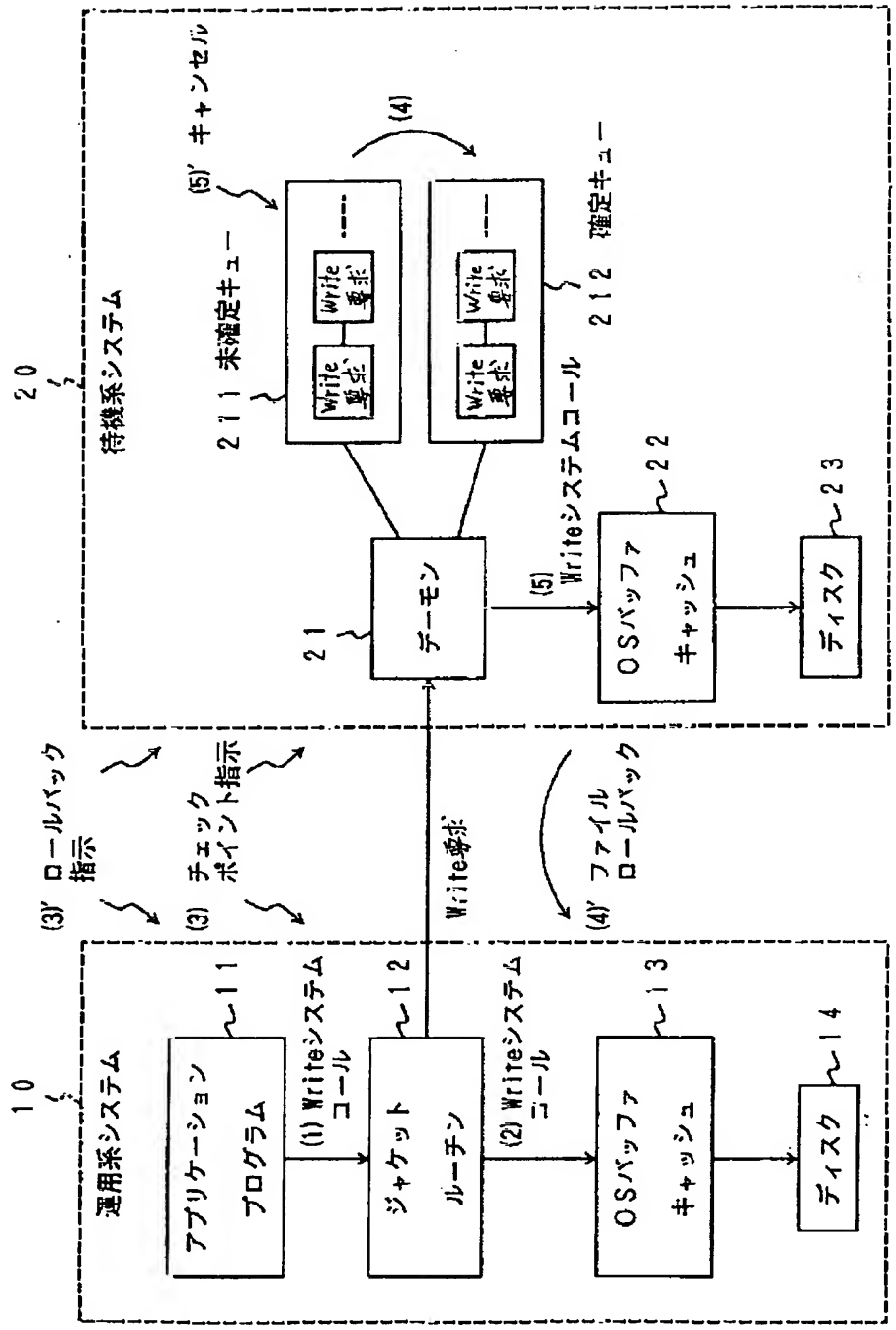
【図７】



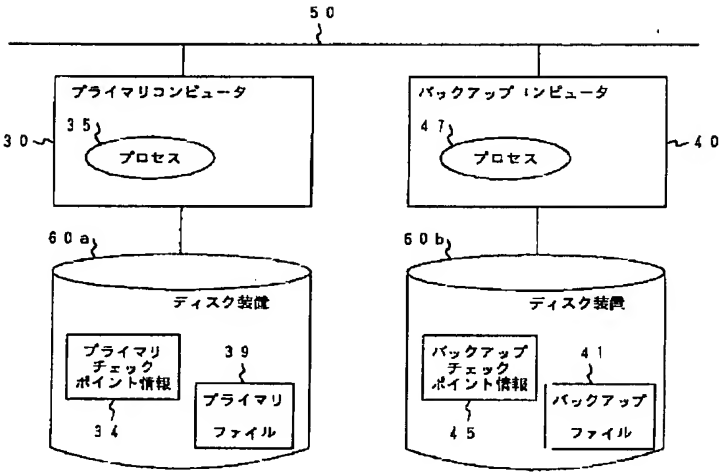
【図９】



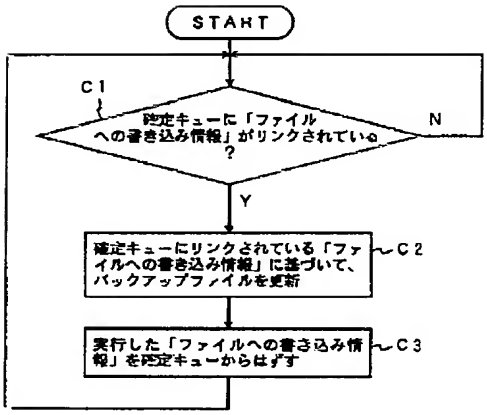
【図1】


20
待機系システム
(5)' キャンセル
211 未確定キュー
Write要求'
Write要求'
(4)
Write要求'
Write要求'
212 確定キュー
21
デーモン
(5) Writeシステムコール
22
OSバッファ
キャッシュ
23
ディスク
(3)' ロールバック指示
(3) チェックポイント指示
Write要求
(4)' ファイルロールバック
10
運用系システム
11
アプリケーション
プログラム
(1) Writeシステムコール
12
ジャケット
ルーチン
(2) Writeシステムコール
13
OSバッファ
キャッシュ
14
ディスク

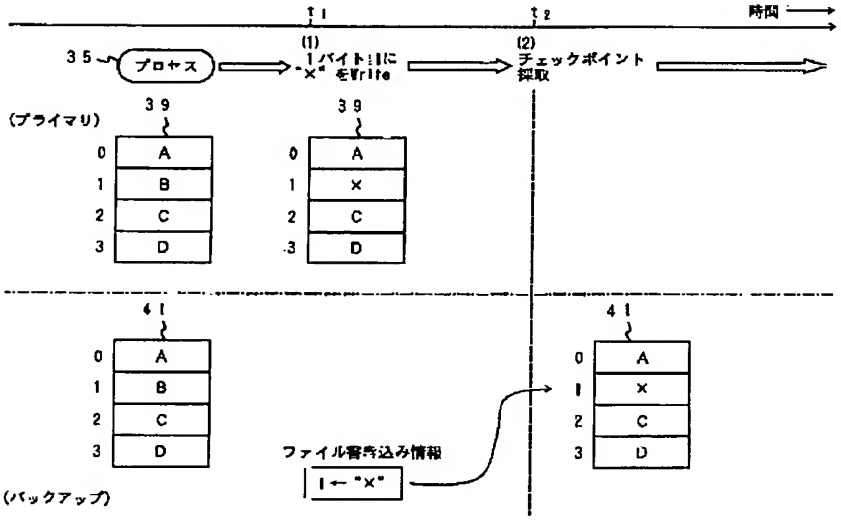
【図３】


50
プライマリコンピュータ
35
プロセス
30
バックアップコンピュータ
47
プロセス
40
60a
ディスク装置
プライマリ
チェック
ポイント情報
34
39
プライマリ
ファイル
60b
ディスク装置
バックアップ
チェック
ポイント情報
45
41
バックアップ
ファイル

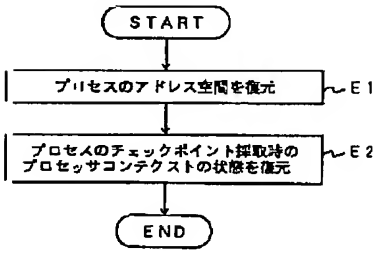
【図８】


START
C1
確定キューに「ファイルへの書き込み情報」がリンクされている？
N
Y
確定キューにリンクされている「ファイルへの書き込み情報」に基づいて、バックアップファイルを更新
C2
実行した「ファイルへの書き込み情報」を確定キューからはずす
C3

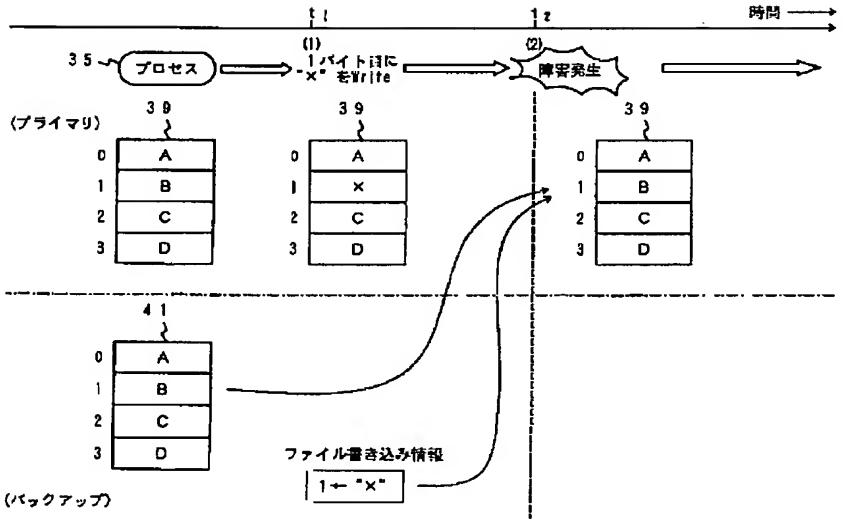
【図４】


t1
t2
時間
35
プロセス
(1)
1バイト目に"x"をWrite
(2)
チェックポイント採取
(プライマリ)
39
0 A
1 B
2 C
3 D
39
0 A
1 x
2 C
3 D
41
0 A
1 B
2 C
3 D
ファイル書き込み情報
1←"x"
41
0 A
1 x
2 C
3 D
(バックアップ)

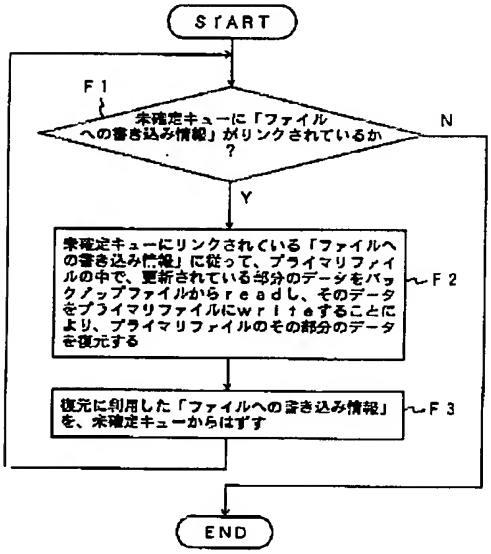
【図１０】


START
プロセスのアドレス空間を復元
E1
プロセスのチェックポイント採取時のプロセッサコンテクストの状態を復元
E2
END

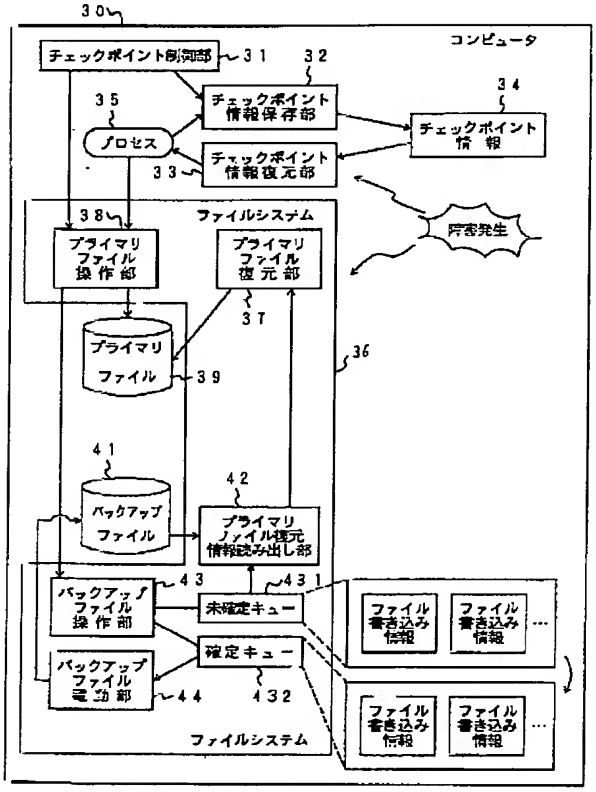
【図５】


t1
t2
時間
35
プロセス
(1)
1バイト目に"x"をWrite
(2)
障害発生
(プライマリ)
39
0 A
1 B
2 C
3 D
39
0 A
1 x
2 C
3 D
39
0 A
1 B
2 C
3 D
41
0 A
1 B
2 C
3 D
(バックアップ)
ファイル書き込み情報
1← "x"

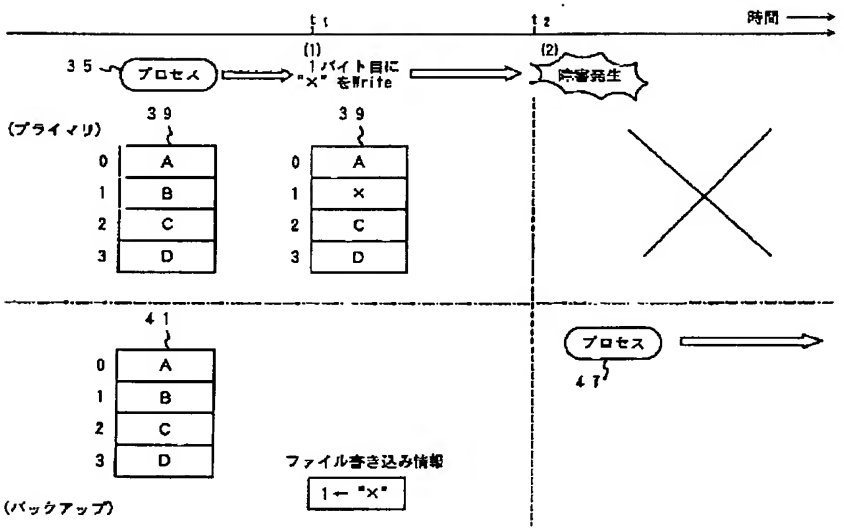
【図１１】


START
F1
未確定キューに「ファイルへの書き込み情報」がリンクされているか？
N
Y
未確定キューにリンクされている「ファイルへの書き込み情報」に従って、プライマリファイルの中で、更新されている部分のデータをバックアップファイルからｒｅａｄし、そのデータをプライマリファイルにｗｒｉｔｅすることにより、プライマリファイルのその部分のデータを復元する
F2
復元に利用した「ファイルへの書き込み情報」を、未確定キューからはずす
F3
END

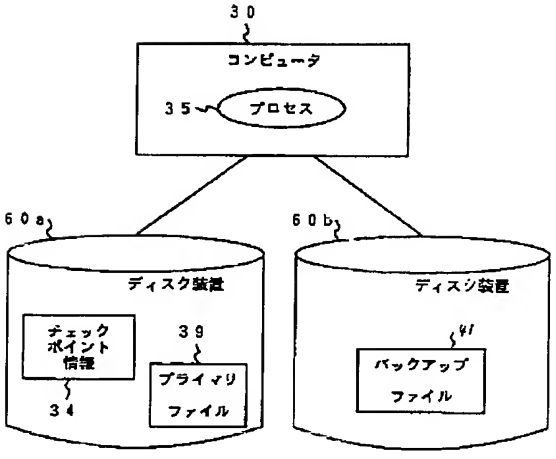
【図１３】


30
コンピュータ
チェックポイント制御部
31
32
チェックポイント情報保存部
34
チェックポイント情報
35
プロセス
チェックポイント情報復元部
33
ファイルシステム
障害発生
38
プライマリファイル操作部
プライマリファイル復元部
37
プライマリファイル
39
36
41
バックアップファイル
42
プライマリファイル復元情報読み出し部
43
バックアップファイル操作部
未確定キュー
431
確定キュー
バックアップファイル管理部
44
432
ファイル書き込み情報
ファイル書き込み情報
ファイル書き込み情報
ファイル書き込み情報
ファイルシステム

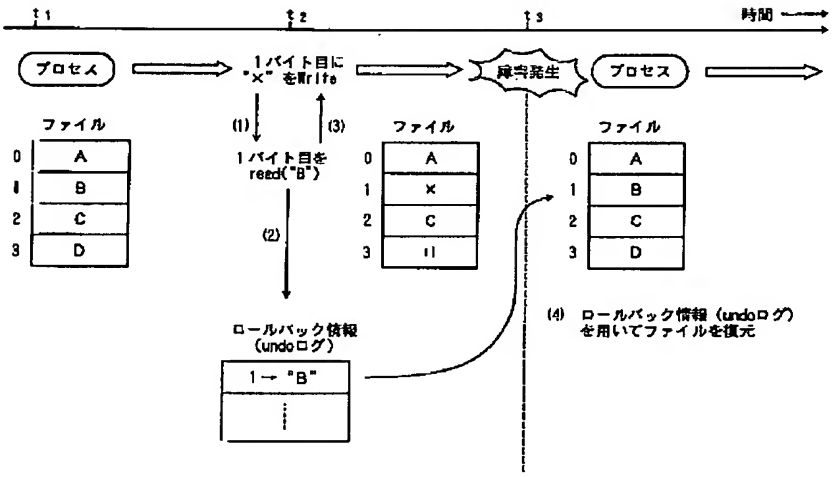
【図１２】


t1
t2
時間
35
プロセス
(1)
1バイト目に
"×" をWrite
(2)
障害発生
39
(プライマリ)
0 A
1 B
2 C
3 D
39
0 A
1 ×
2 C
3 D
41
0 A
1 B
2 C
3 D
(バックアップ)
ファイル書き込み情報
1← "×"
プロセス
47

【図１４】


30
コンピュータ
35
プロセス
60a
ディスク装置
チェック
ポイント
情報
34
39
プライマリ
ファイル
60b
ディスク装置
61
バックアップ
ファイル

【図１５】


t1
t2
t3
時間
プロセス
1バイト目に "×" をWrite
障害発生
プロセス
ファイル
0 A
1 B
2 C
3 D
(1)
(3)
1バイト目を read("B")
(2)
ファイル
0 A
1 ×
2 C
3 ||
ファイル
0 A
1 B
2 C
3 D
ロールバック情報 (undoログ)
1→ "B"
(4) ロールバック情報（undoログ）を用いてファイルを復元